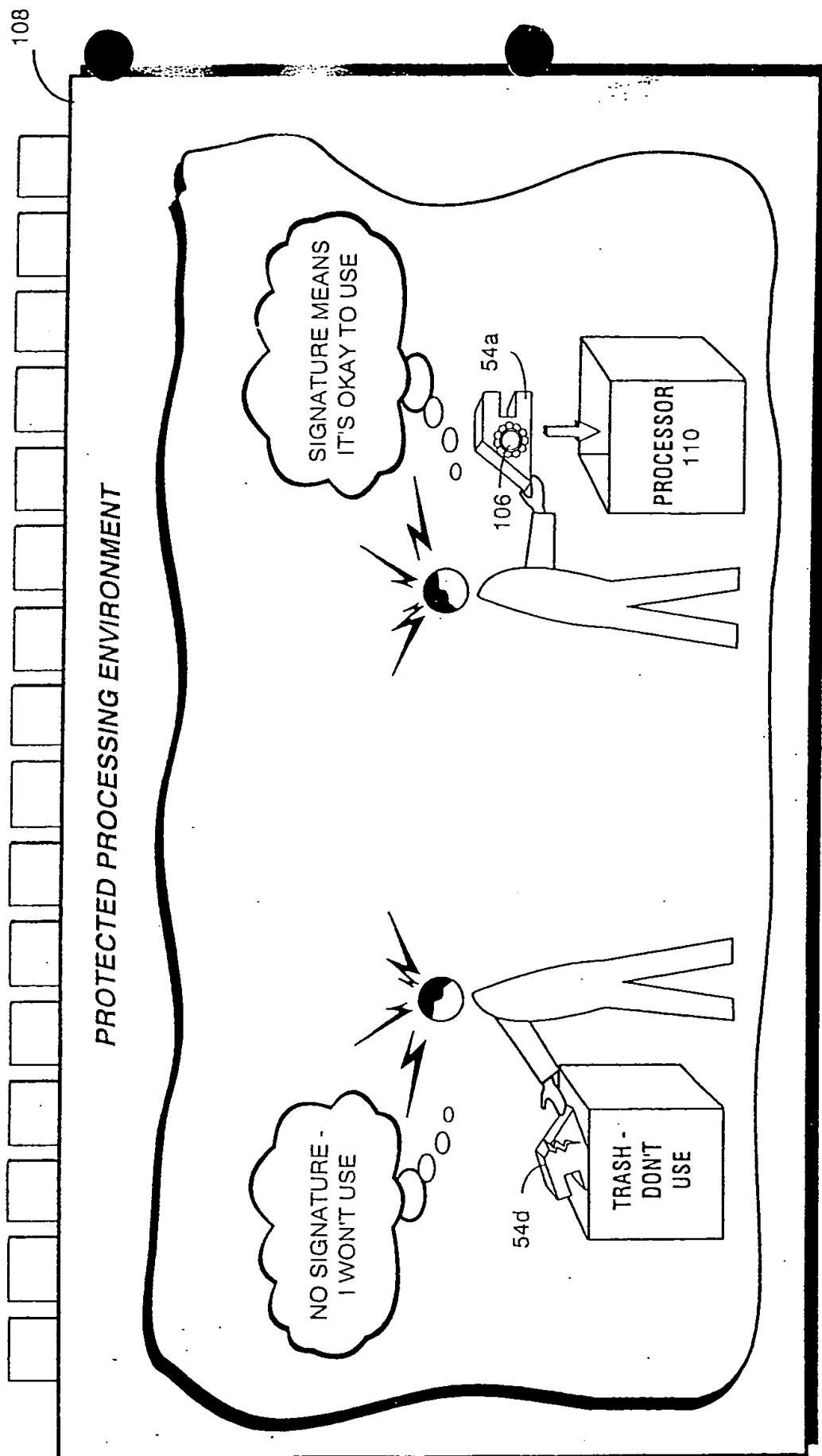


FIG. 3 Before Protected Processing Environment Uses A Load Module, It Checks To See If Load Module Has Been Verified



008270" 26982960

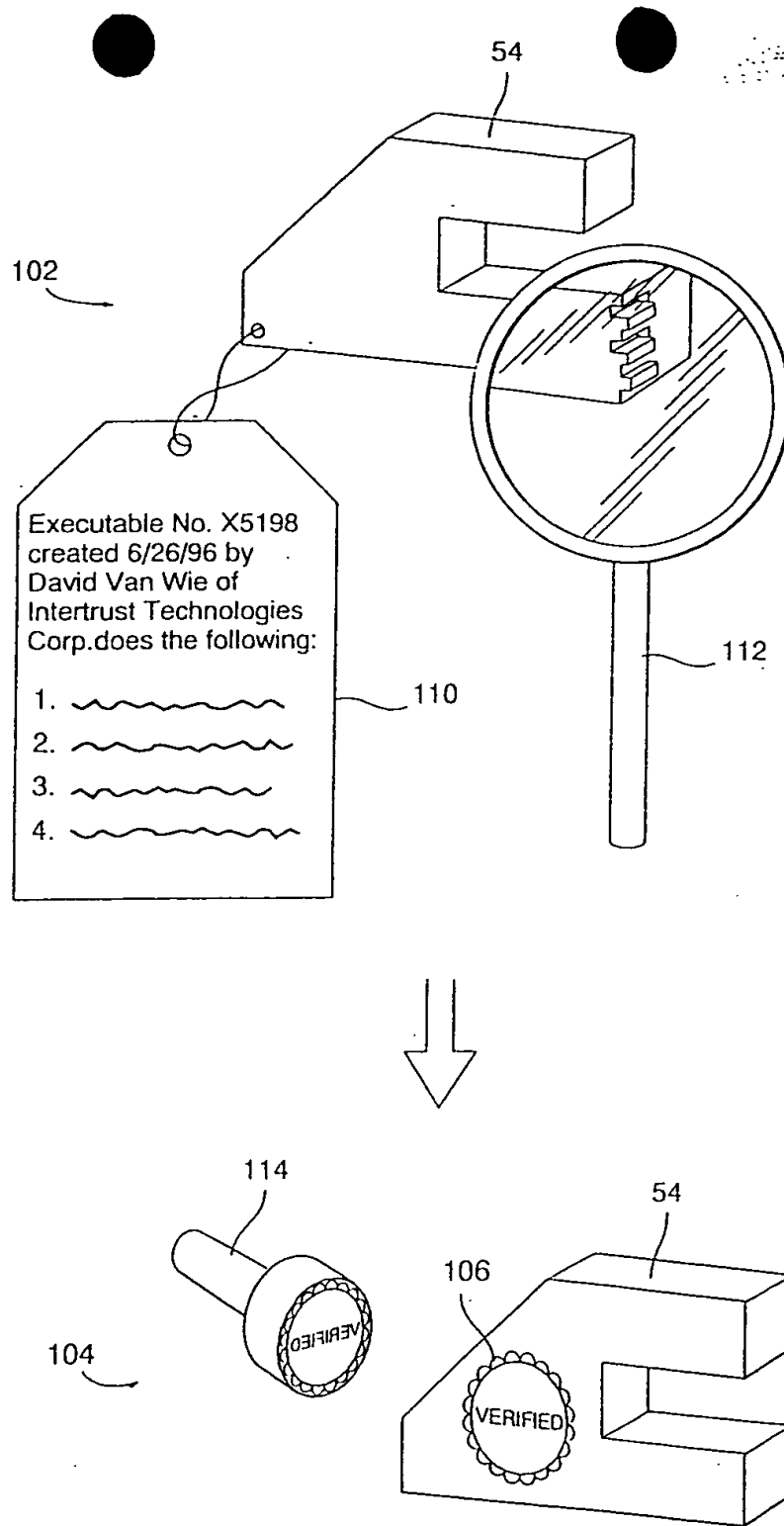


FIG. 4

**Certifying Load Module by
Checking it Against its Documentation**

008270" 26982950

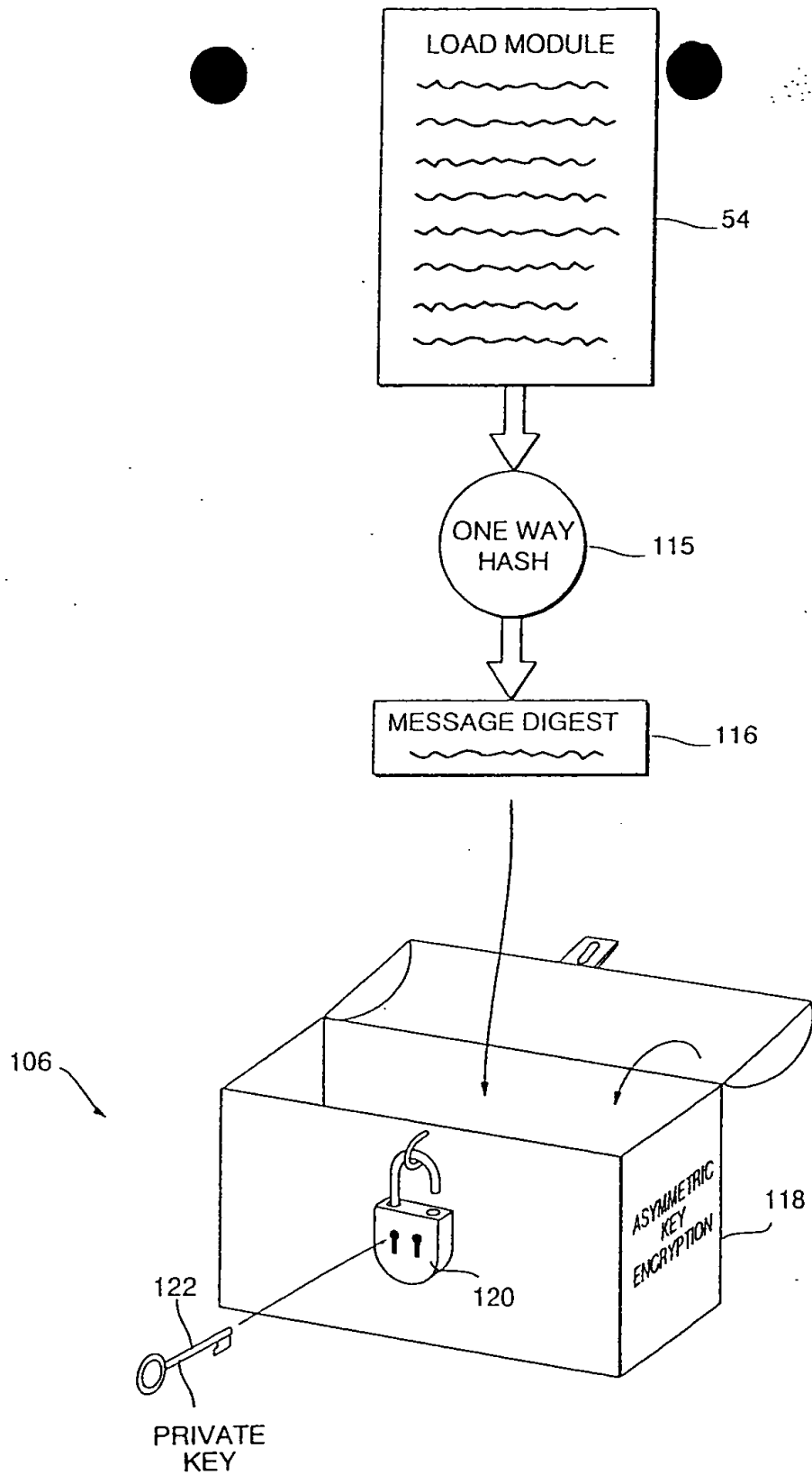


FIG. 5
Creating a Certifying
Digital Signature

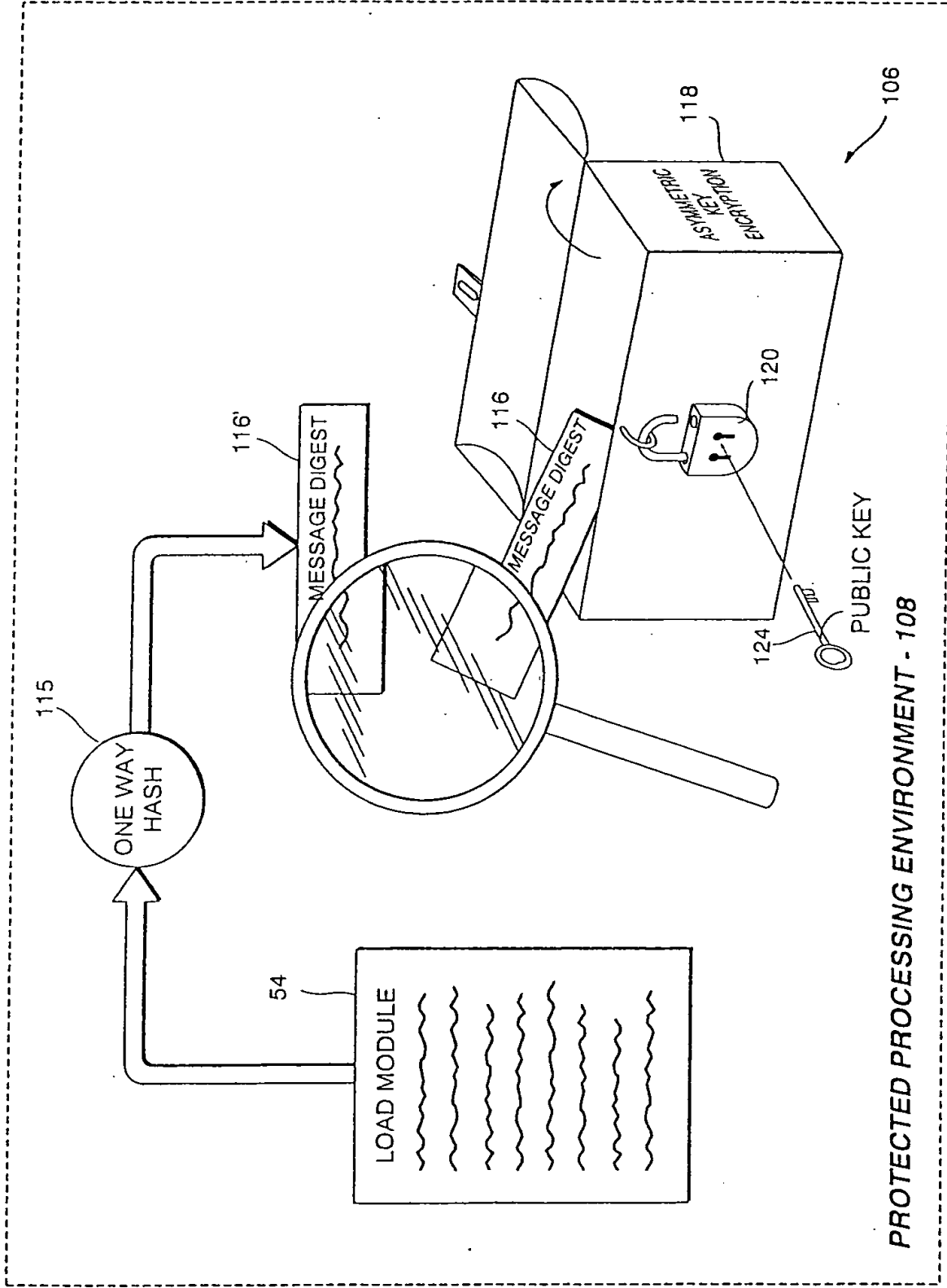


FIG. 6 Authenticating a Digital Signature

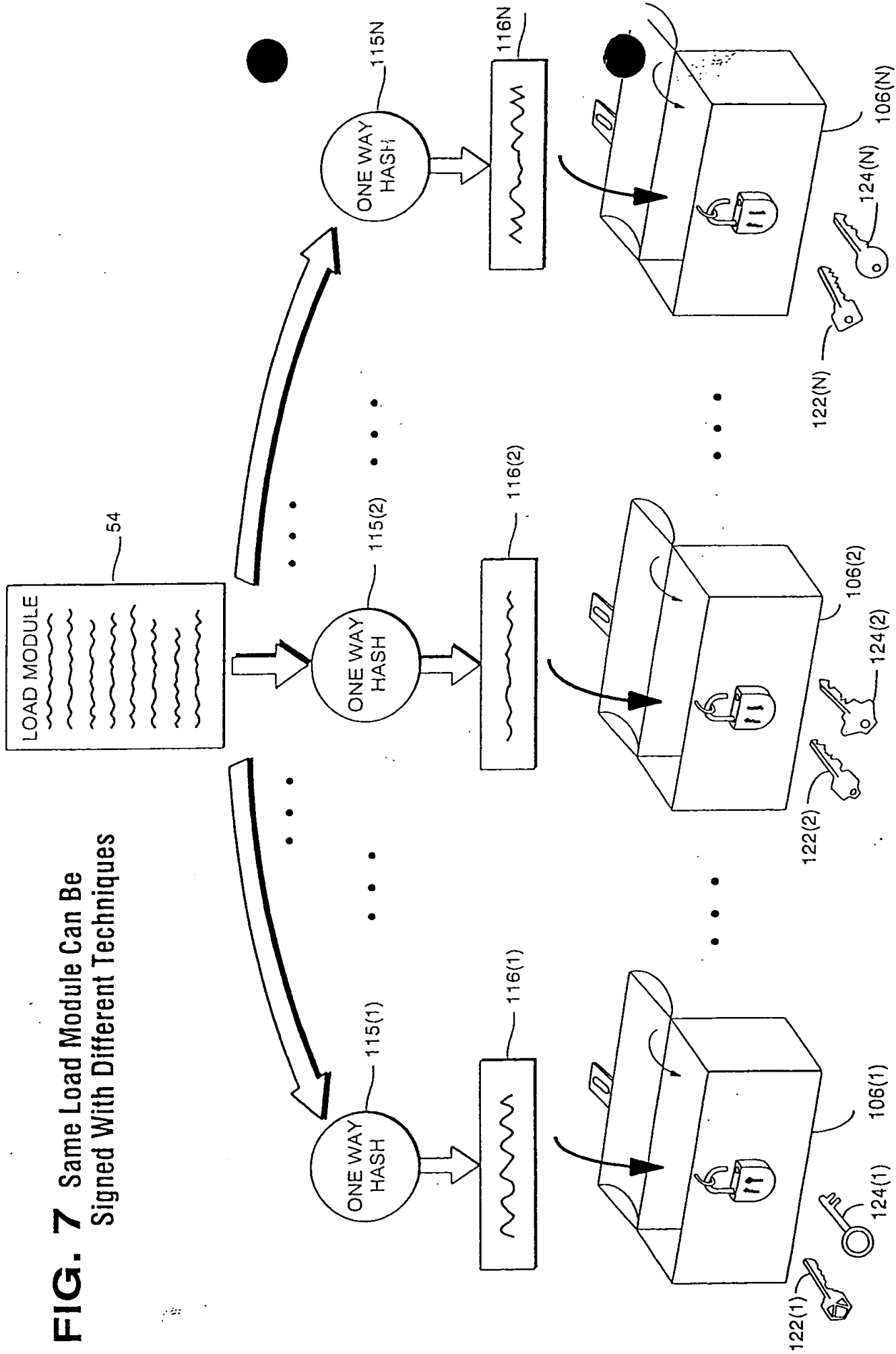


FIG. 7 Same Load Module Can Be Signed With Different Techniques

THE UNIVERSITY OF CHICAGO

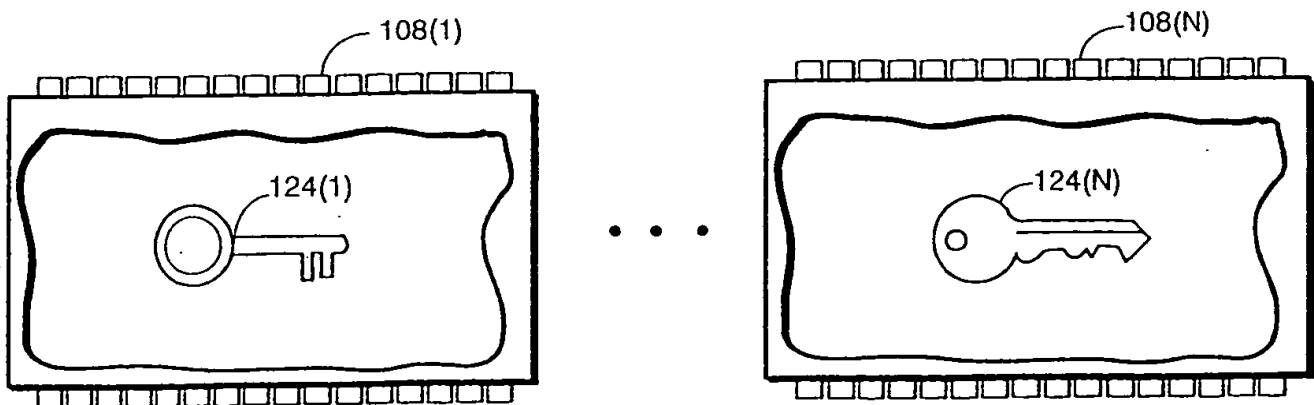
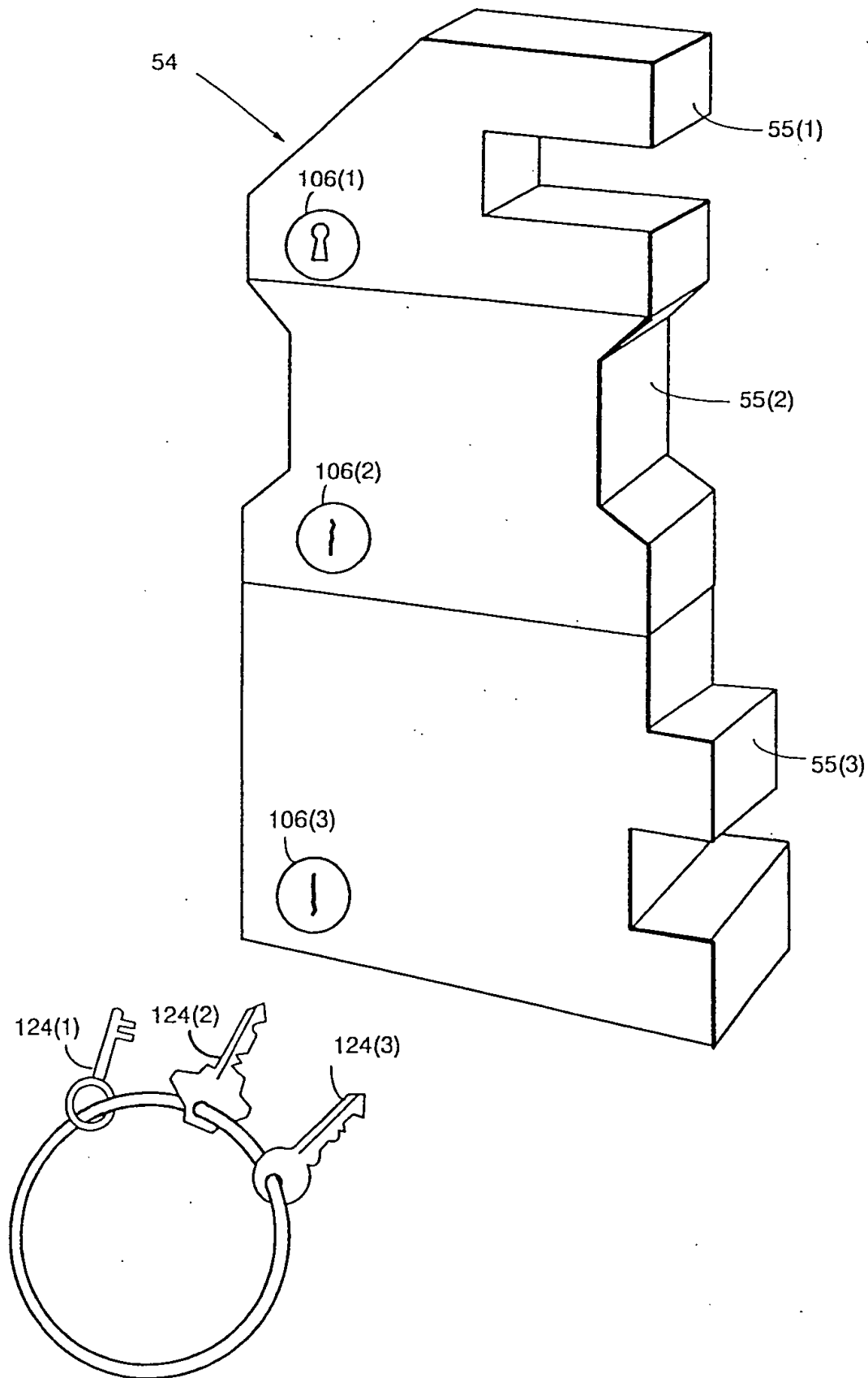


FIG. 9 Load Module Can Have Several Independently Signed Portions



008220" 26982960

FIG. 10A Assurance Level I
Software-Based
Protected Processing Environment

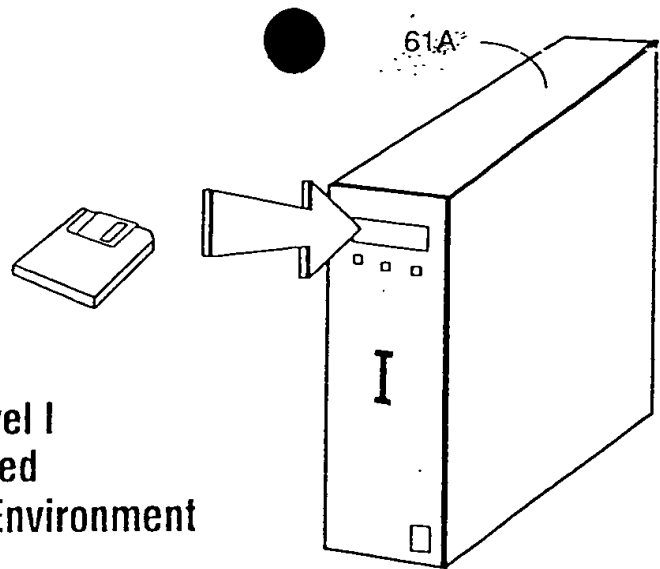


FIG. 10B Assurance Level II
Software and Hardware-Based
Protected Processing Environment

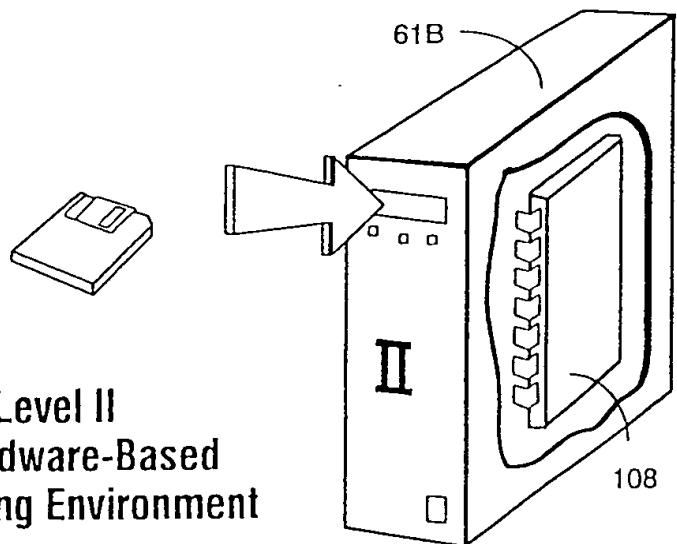
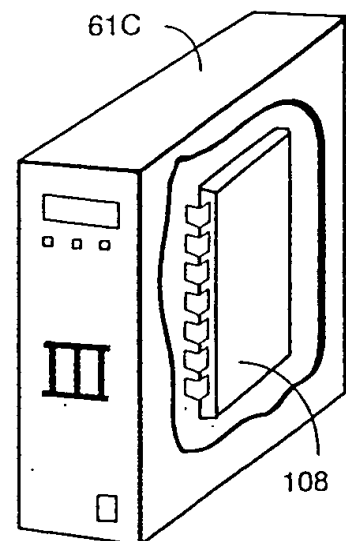
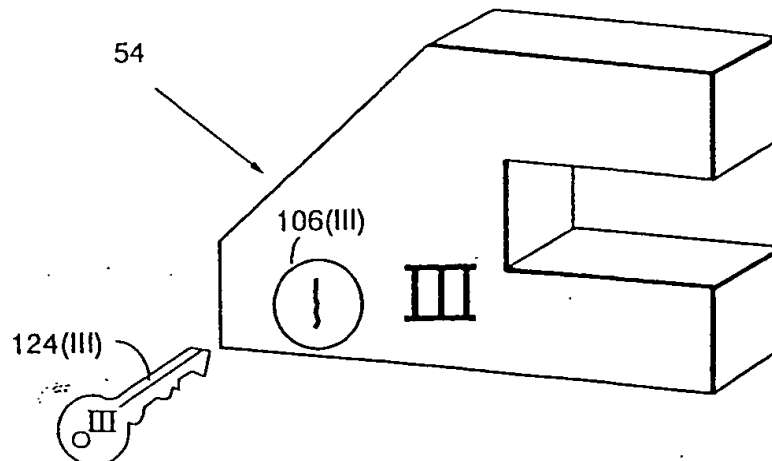


FIG. 10C Assurance Level III
Hardware-Based
Protected Processing Environment





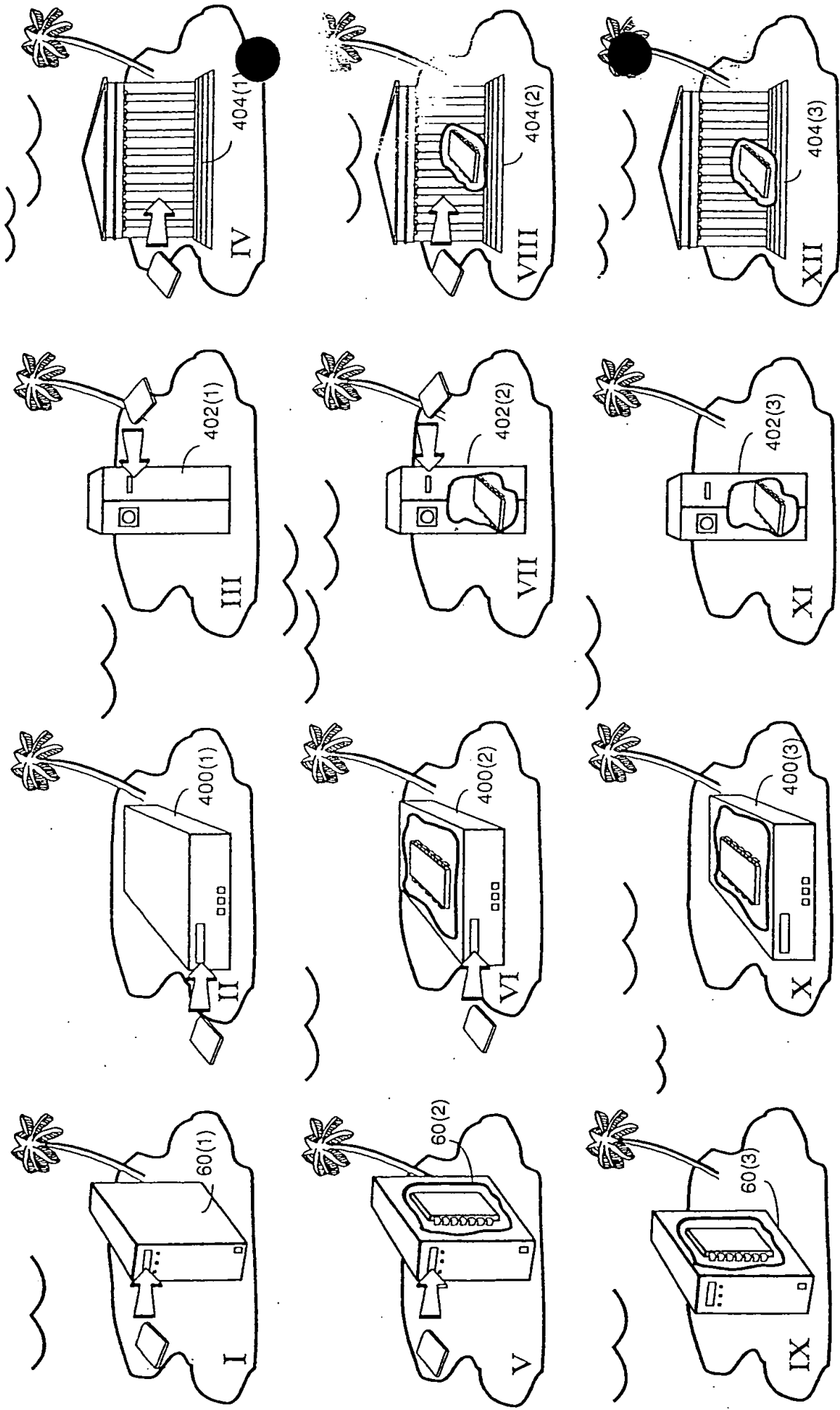


FIG. 12 Using Digital Signatures For Compartmentalizing Different Assurance Levels

FIG. 13 Multiple Assurance Levels

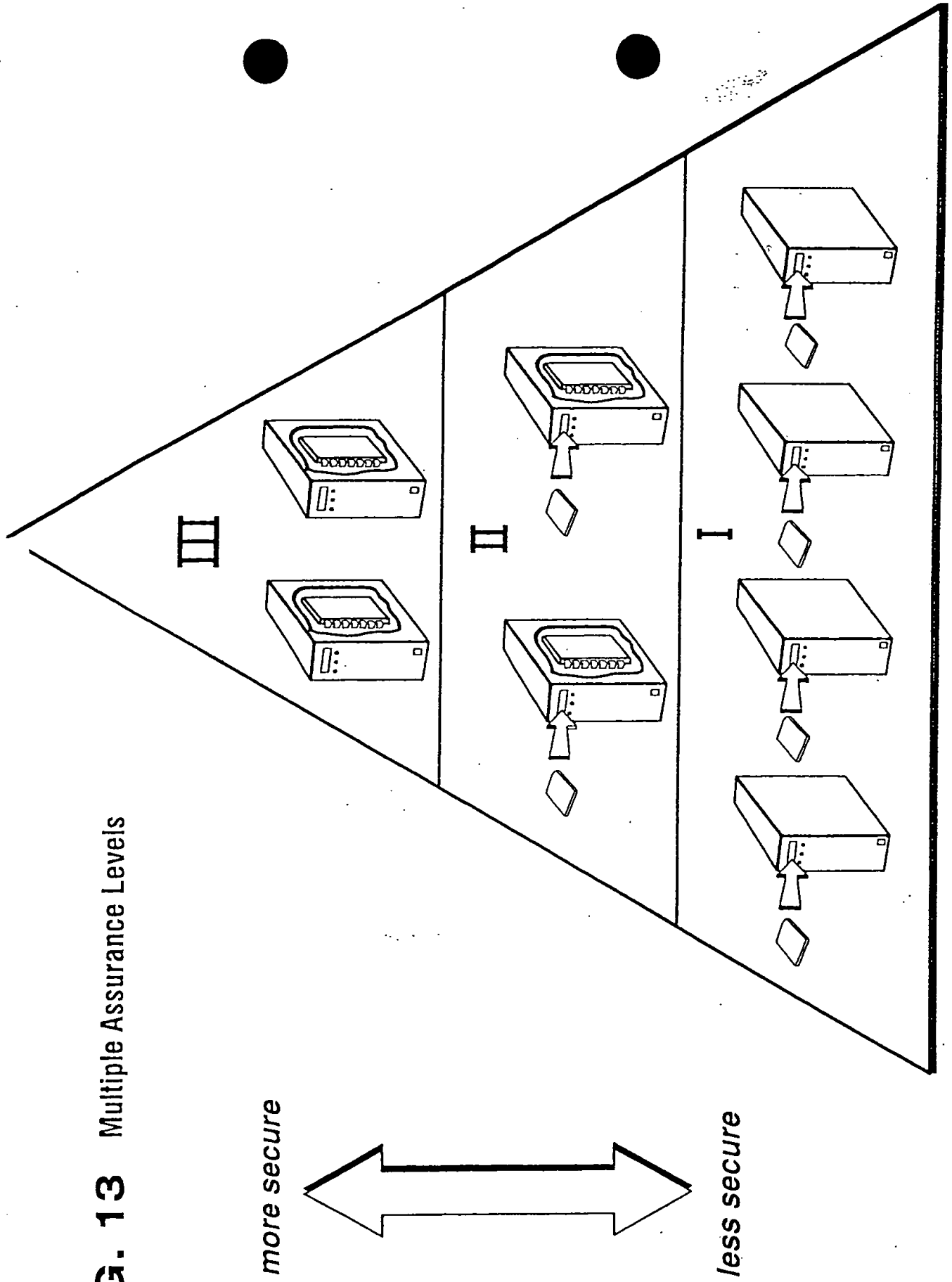
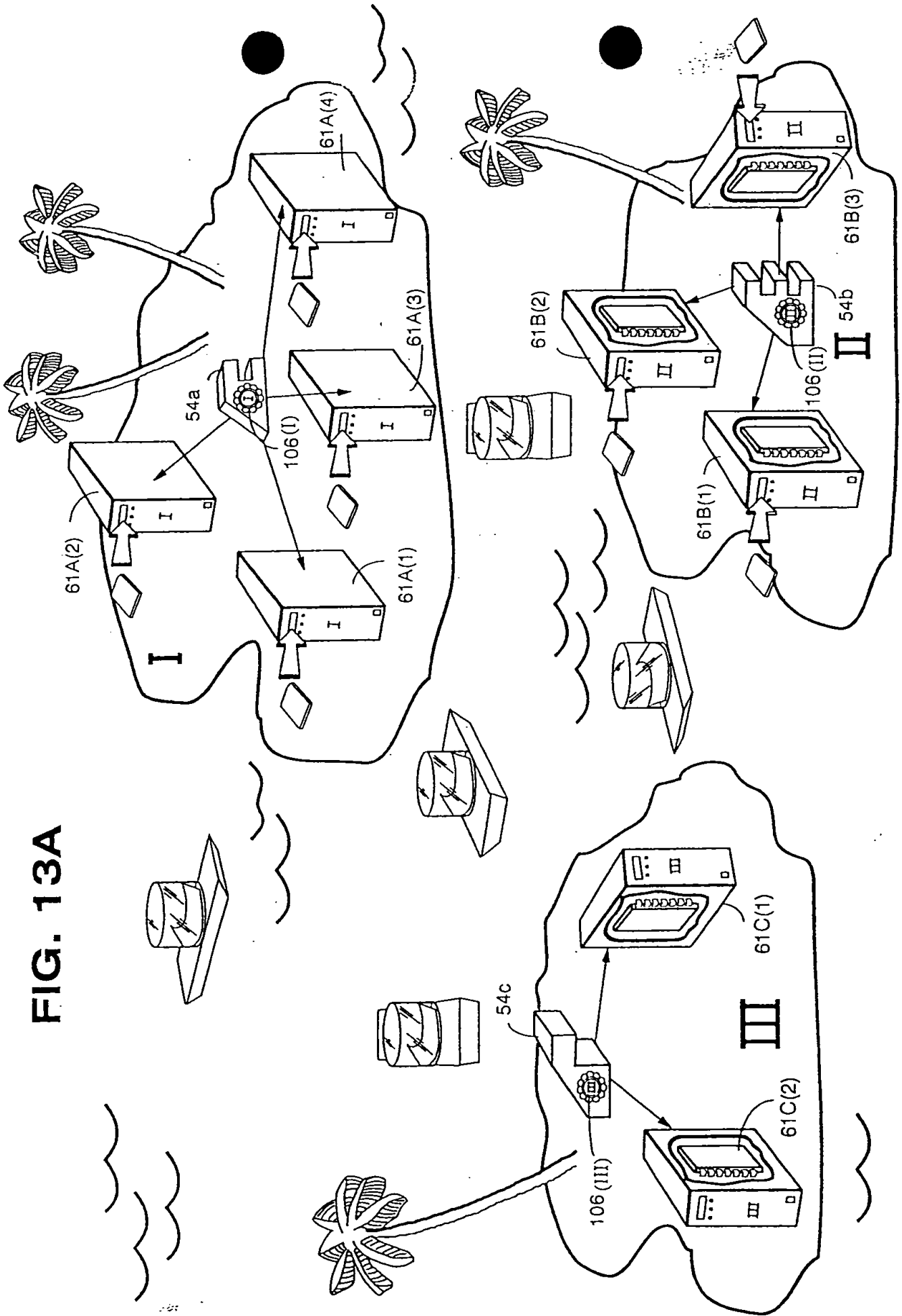
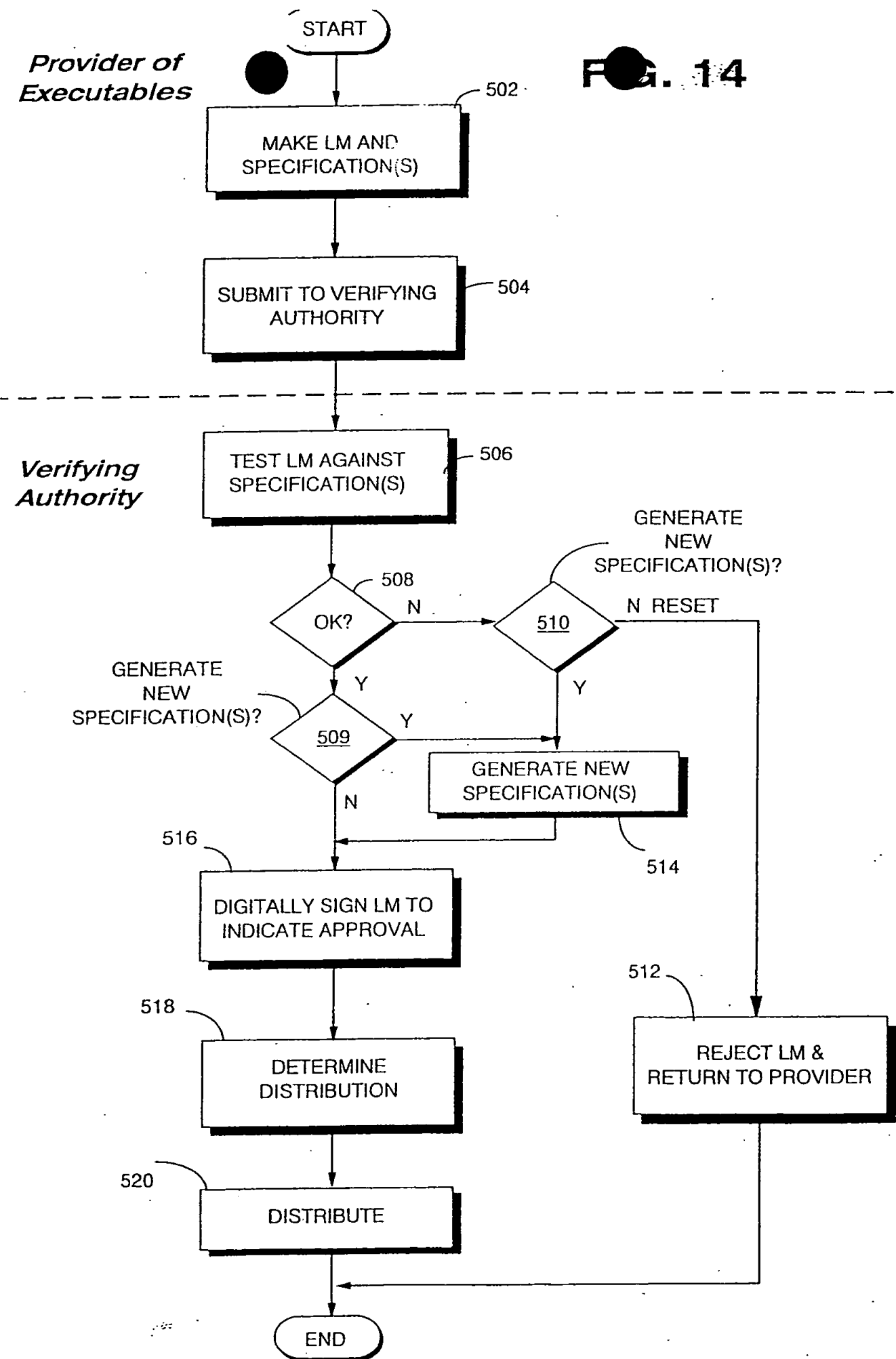


FIG. 13A





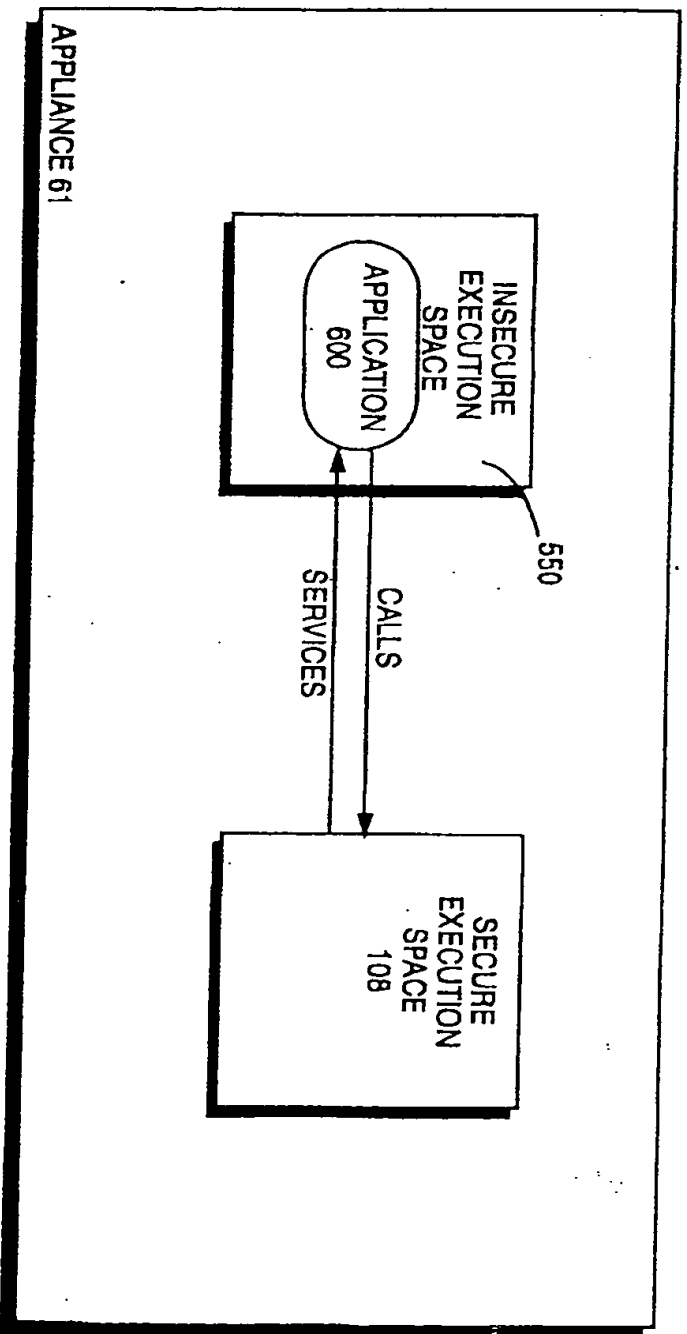


FIG. 15 EXAMPLE APPLIANCE EXECUTING APPLICATION PROGRAM IN INSECURE EXECUTION SPACE

09628592.072800

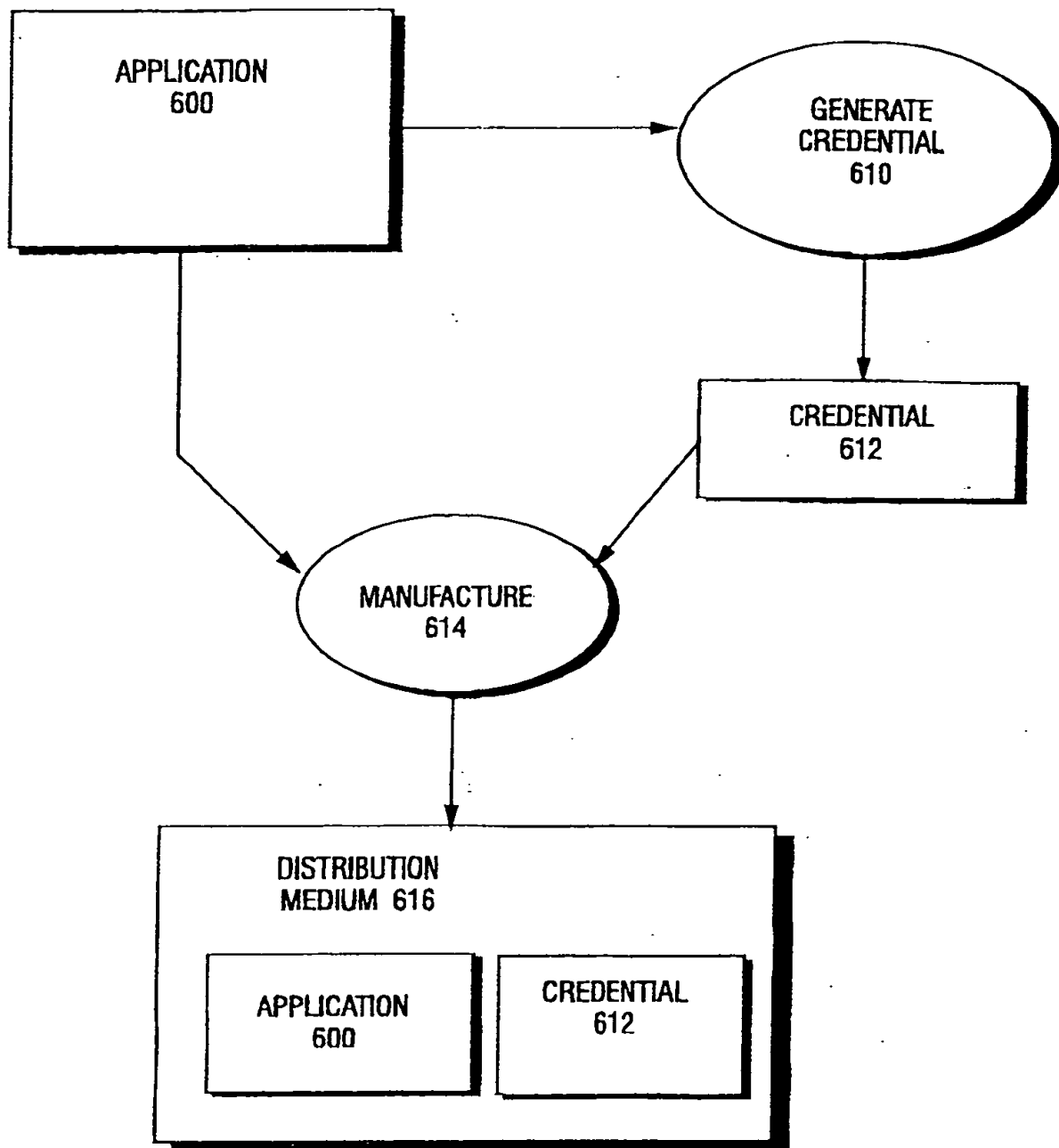


FIG. 16 EXAMPLE APPLICATION CERTIFICATION PROCESS

The diagram illustrates the architecture of an application, labeled 600. It consists of several components arranged within a large rectangular frame:

- READ-ONLY COMPONENT 601(N)**: Located in the top-left corner.
- READ-WRITE COMPONENT 603(1)**: Located in the top-right corner.
- READ-WRITE COMPONENT 603(N)**: Located in the center of the diagram.
- EXECUTABLE COMPONENT 601(1)**: Located in the bottom-left corner.
- LIBRARY COMPONENT 601(2)**: Located in the bottom-right corner.

Four small square markers are positioned vertically to the right of the central component 603(N). The entire set of components is enclosed within a large rectangle labeled **APPLICATION** at the bottom center, with the reference numeral **600** at the bottom right.

FIG. 16A EXAMPLE APPLICATION PROGRAM AND COMPONENTS

```

graph TD
    610([START]) --> 700[SELECT APPLICATION COMPONENT PORTION]
    700 --> 702[HASH BYTES IN SELECTED PORTION TO YIELD HASH VALUE]
    702 --> 704[GENERATE HASH BLOCK DESCRIBING EACH CALCULATED PORTION HASH VALUE]
    704 --> 708{REQUIRED QUANTITY OF RANGE HASHES CALCULATED?}
    708 -- N (REPEAT) --> 700
    708 -- Y --> 710[HASH SET OF HASH BLOCKS]
    710 --> 712[DIGITALLY SIGN THE HASH]
    712 --> 714[ENCRYPT SET OF HASH BLOCKS AND DIGITIZE SIGNATURE TO CREATE CREDENTIAL PART]
    714 --> 716[COMBINE CREDENTIAL PARTS TO PRODUCE CREDENTIAL]
    716 --> END([END])
  
```

FIG. 17

EXAMPLE CREDENTIAL CREATION PROCESS

008270" 26982960

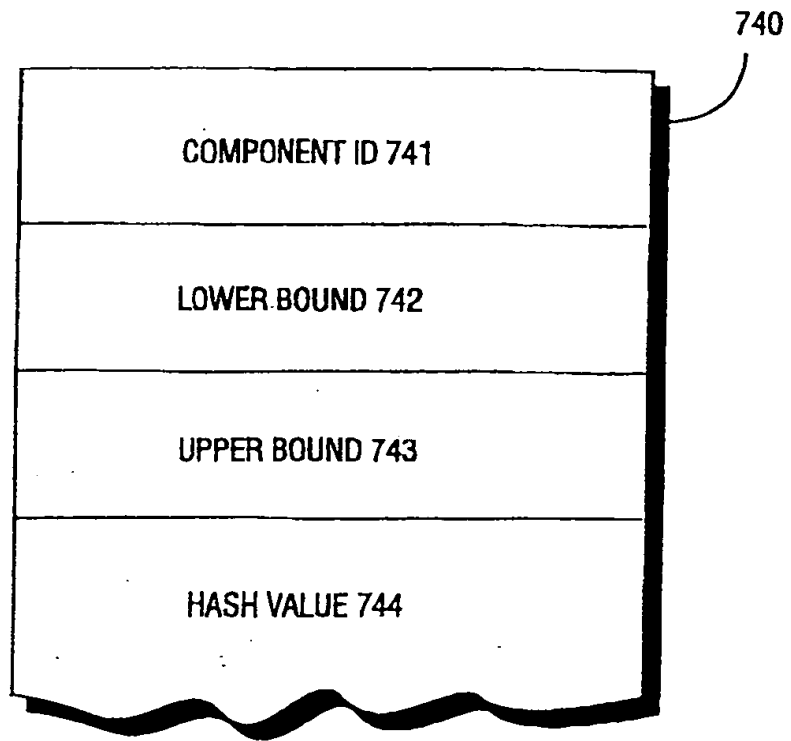


FIG. 18 EXAMPLE HASH BLOCK

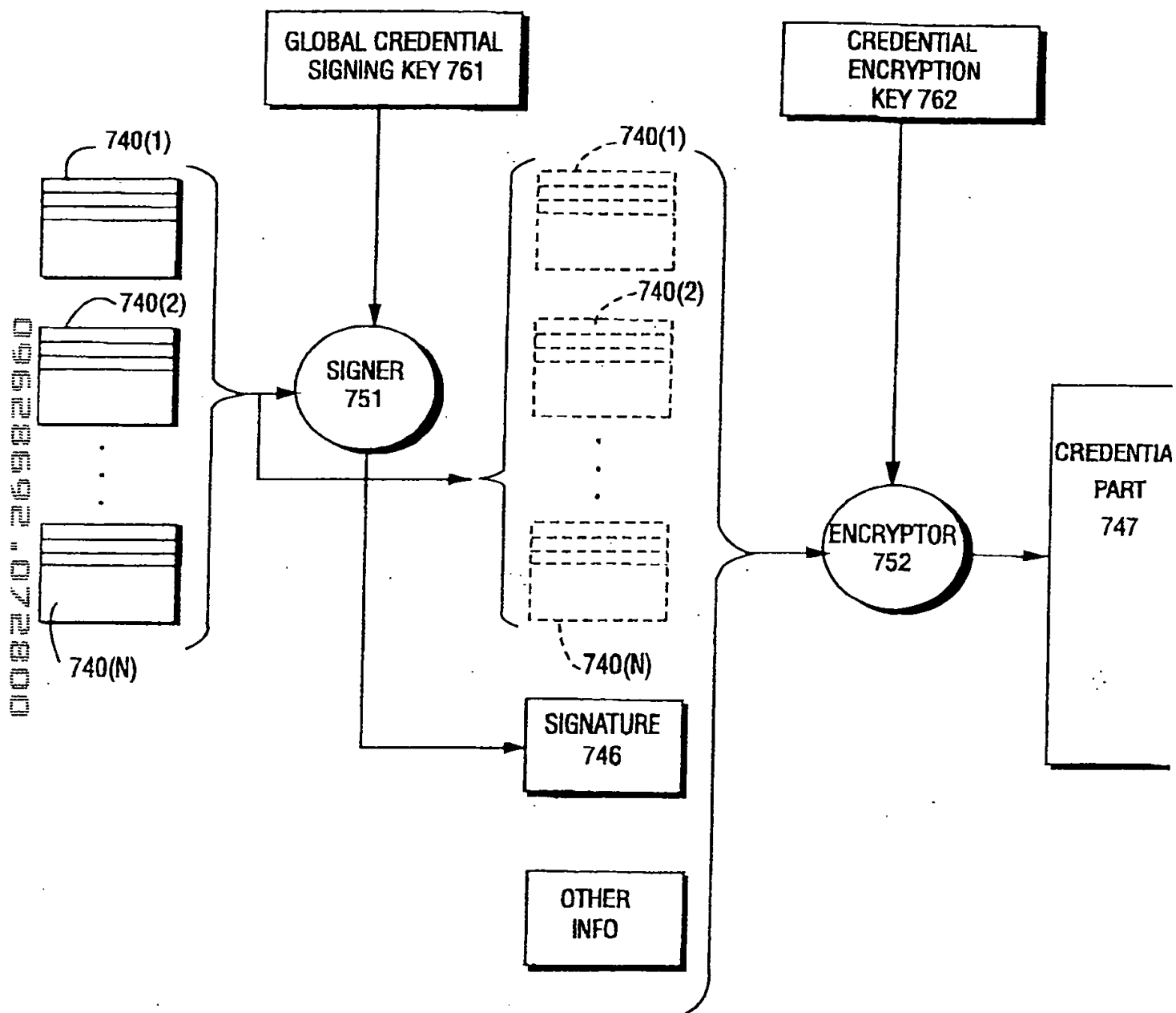


FIG. 19 EXAMPLE CREDENTIAL CREATION

The diagram illustrates a security attack on a validation process. At the top, a large box labeled "APPLICATION 600" contains a smaller box labeled "CRITICAL COMPONENT 804". To the right of the application box, a vertical bracket labeled "VALIDATION RANGES" spans the height of the application box. This bracket is divided into four segments, each with a horizontal line pointing to a label: "OK", "OK", "FAIL", and "FAIL". Below the application box, a box labeled "ATTACKER'S CRITICAL COMPONENT 802" is shown. An arrow labeled "ATTACKER SUBSTITUTES" points from the attacker's component box to the critical component box (804) inside the application box. To the right of the attacker's component box, a box labeled "64" is shown, representing the attacker. The attacker is depicted as a figure in a top hat and long coat, holding a cane, with an arrow pointing from the label "64" to the figure.

• 45

008270" 26982960

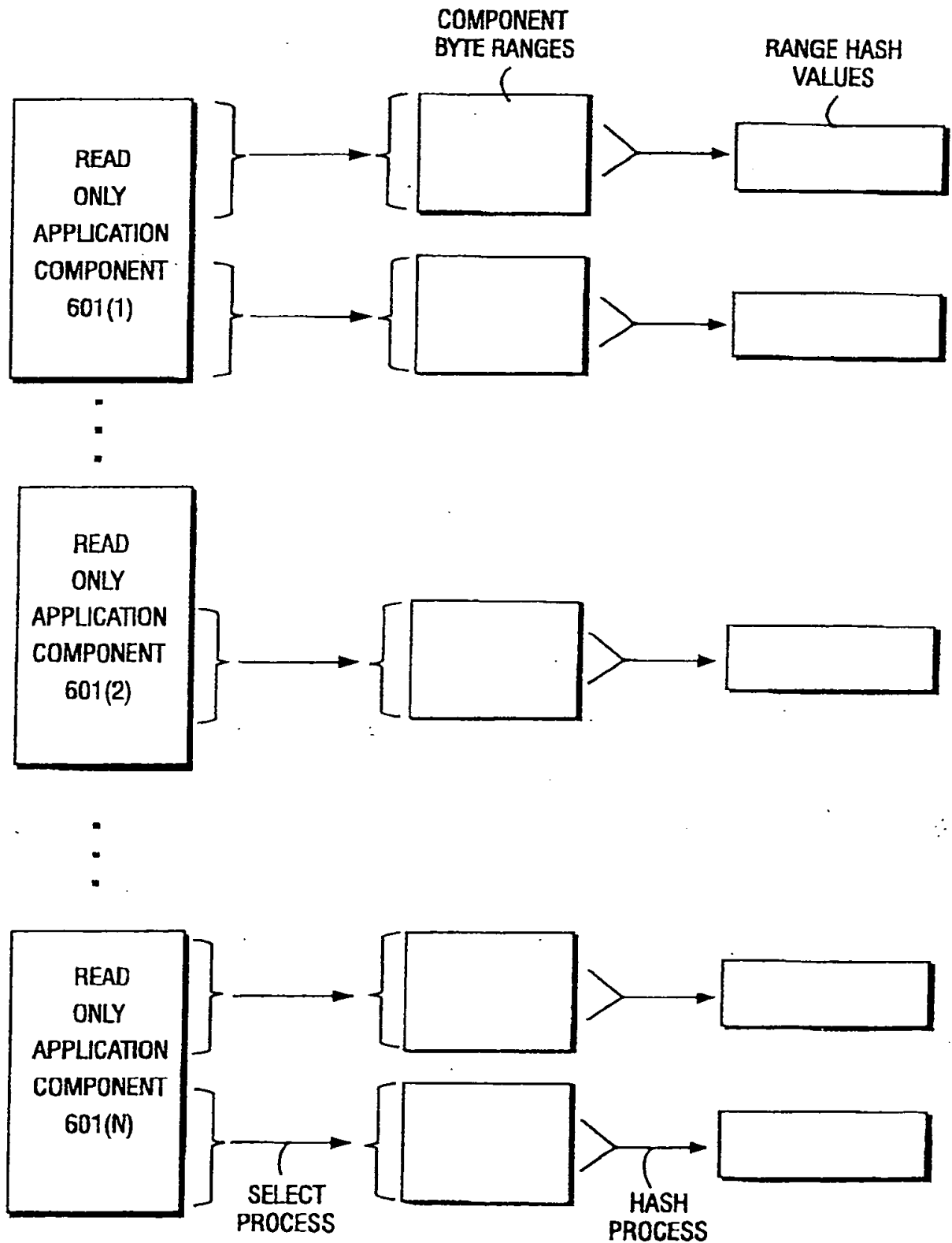


FIG. 20A EXAMPLE NON-OVERLAPPING HASH RANGES

008220" 26982960

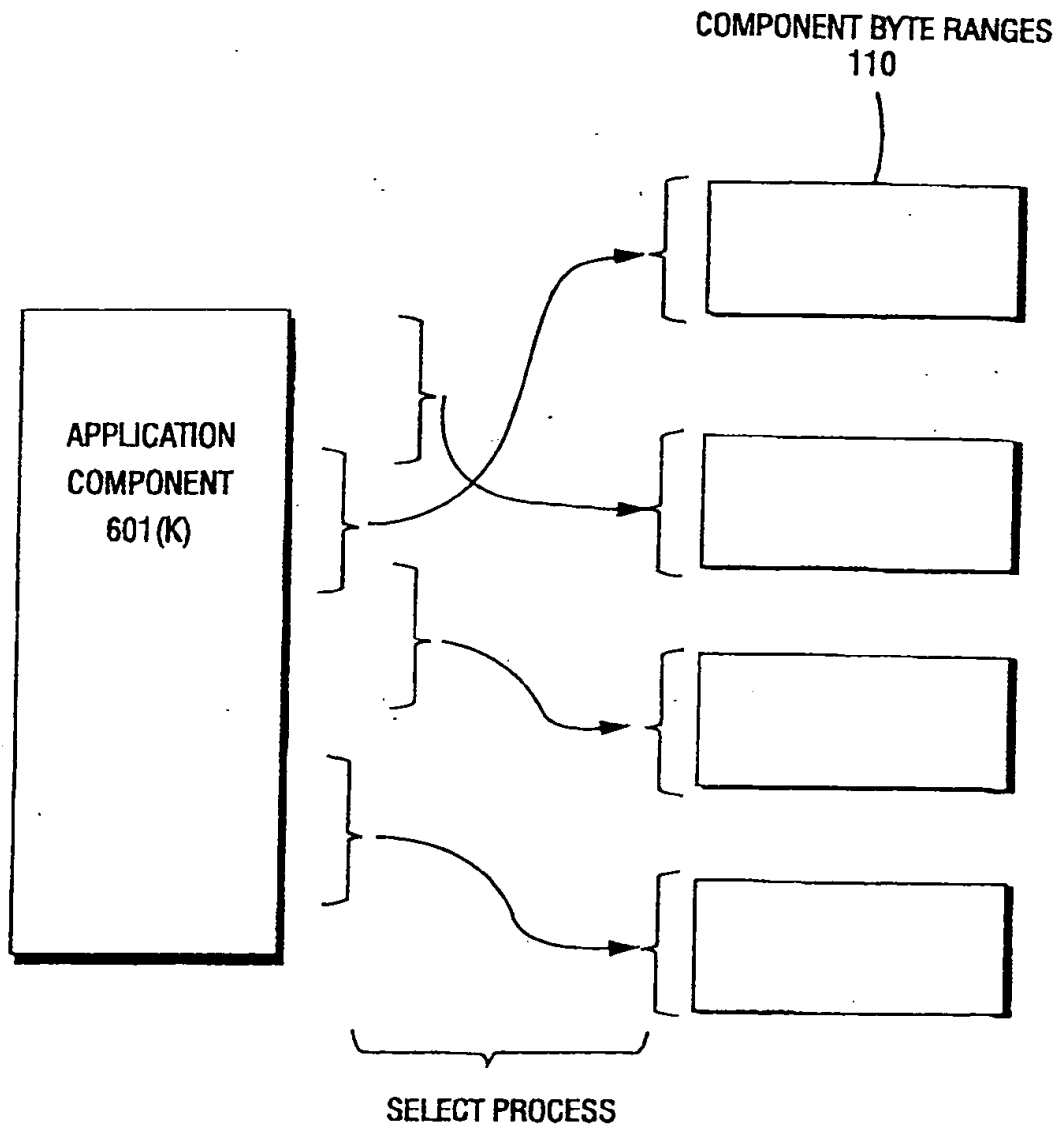


FIG. 20B EXAMPLE OF OVERLAPPING HASH RANGES

Diagram illustrating the structure of Application 600, showing a central **CRITICAL COMPONENT** box and multiple paths (PATH 1, PATH 2, PATH N) connecting to it.

The diagram shows the following components and connections:

- CRITICAL COMPONENT**: A central rectangular box containing three smaller rectangular blocks.
- PATH 1**: A path starting from a rectangular block at the top left, labeled **850(1)**, which connects to the top-left block inside the **CRITICAL COMPONENT**.
- PATH 2**: A path starting from a rectangular block at the top right, labeled **850(2)**, which connects to the top-right block inside the **CRITICAL COMPONENT**.
- PATH N**: A path starting from a rectangular block at the bottom right, labeled **850(N)**, which connects to the bottom-right block inside the **CRITICAL COMPONENT**.
- 80A**: A label indicating a specific point or connection near the top of the **CRITICAL COMPONENT**.

FIG. 20C PSEUDO-RANDOM VALIDATION PATHS IN APPLICATION

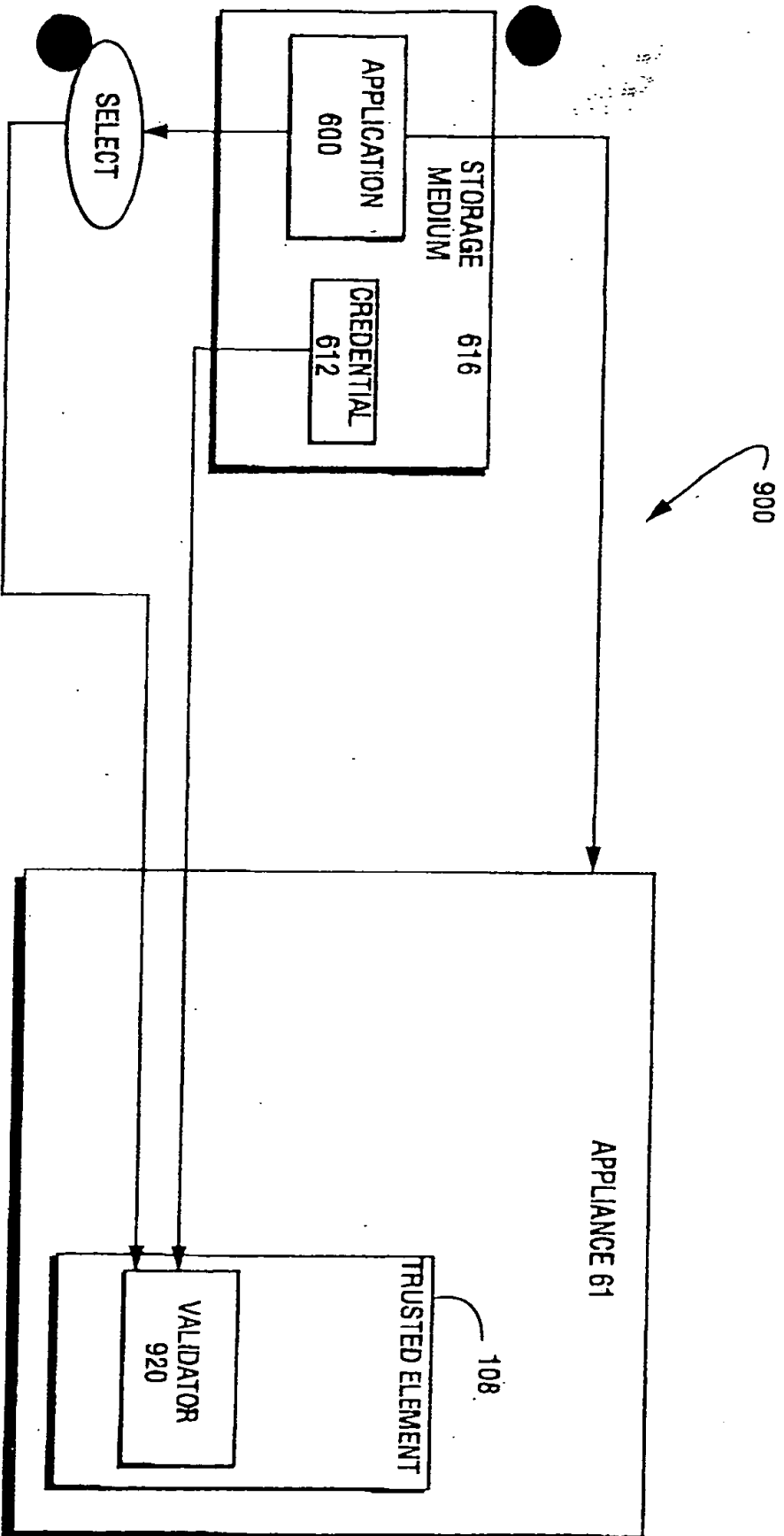
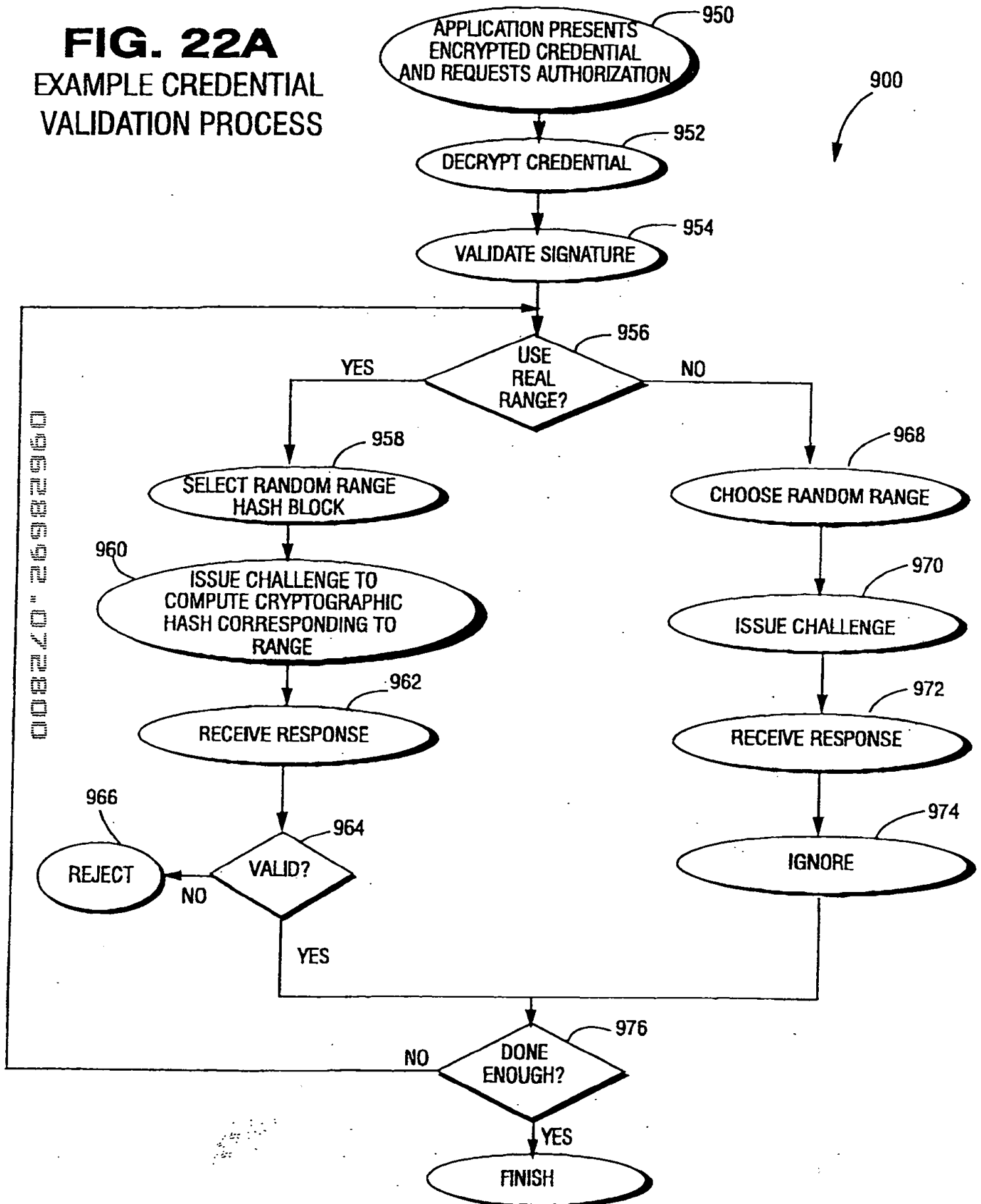


FIG. 21 EXAMPLE CREDENTIAL VALIDATION PROCESS

FIG. 22A
EXAMPLE CREDENTIAL
VALIDATION PROCESS



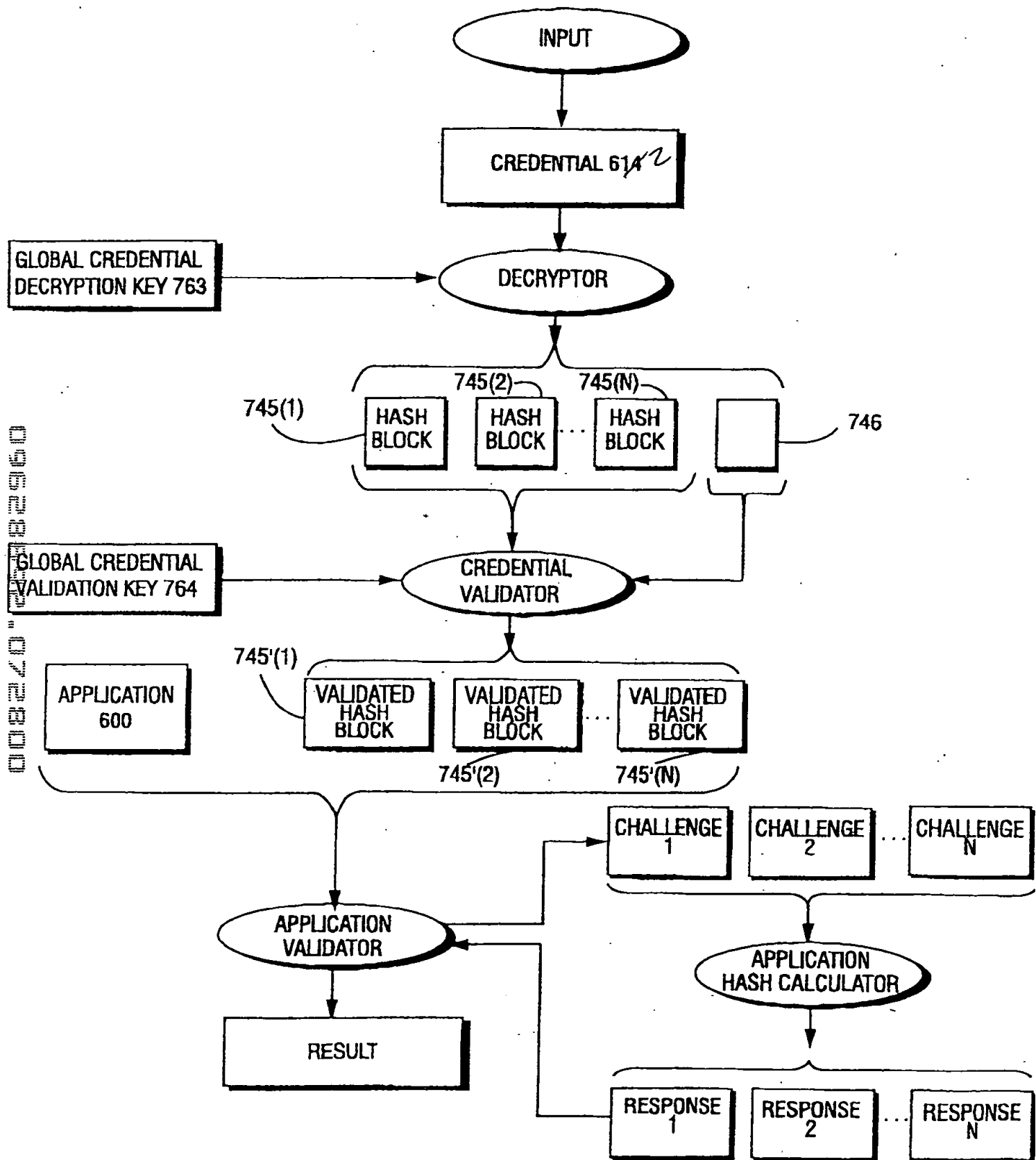


FIG. 22B EXAMPLE CREDENTIAL VALIDATION